# Tiny Transformers: Enabling Transformer Execution on Low-Power IoT Endnodes

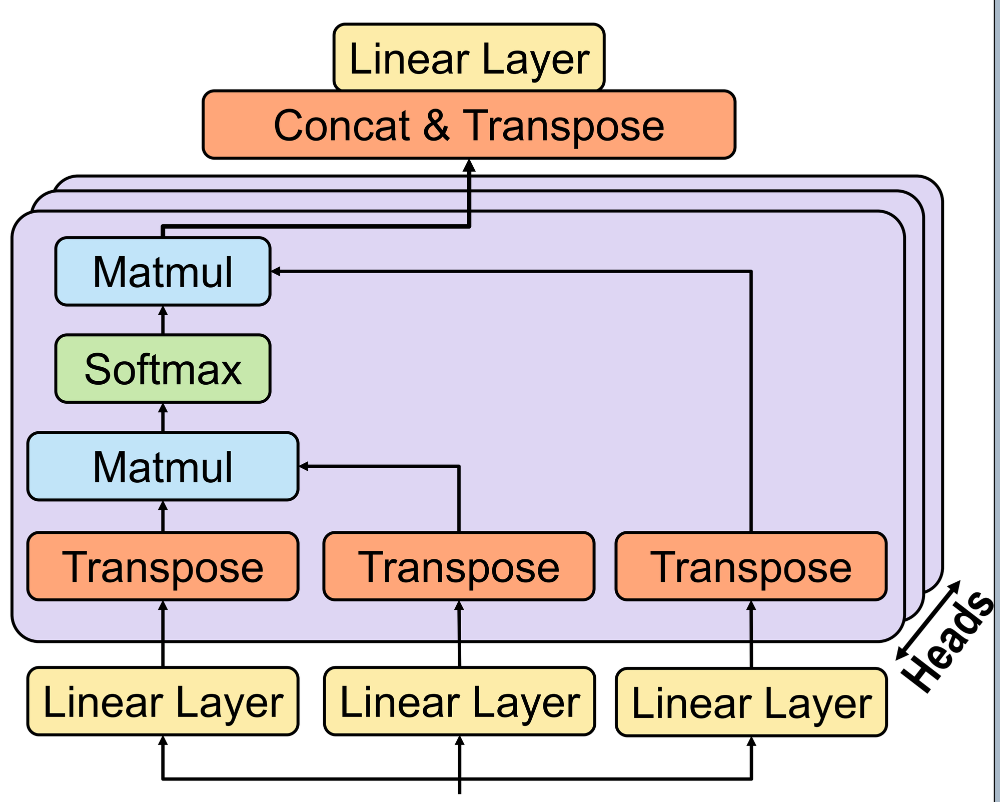
**Moritz Scherer**[1], Alessio Burrello[2], Marcello Zanghieri[2], Luca Benini[1,2], and Francesco Conti[2]
[1]ETH Zurich, [2]University of Bologna

## Self-Attention

Transformers are SoA in many fields
- Computer Vision, Audio, NLP, many more

**Self-attention is key layer in transformers**
- **Linear layers ➡ many parameters**
- Non-trivial data dependencies

**Multihead Self-Attention**




Deploying self-attention to MCUs is challenging
- Typically many parameters
- Quantization of Softmax
- **No efficient open-source kernels**

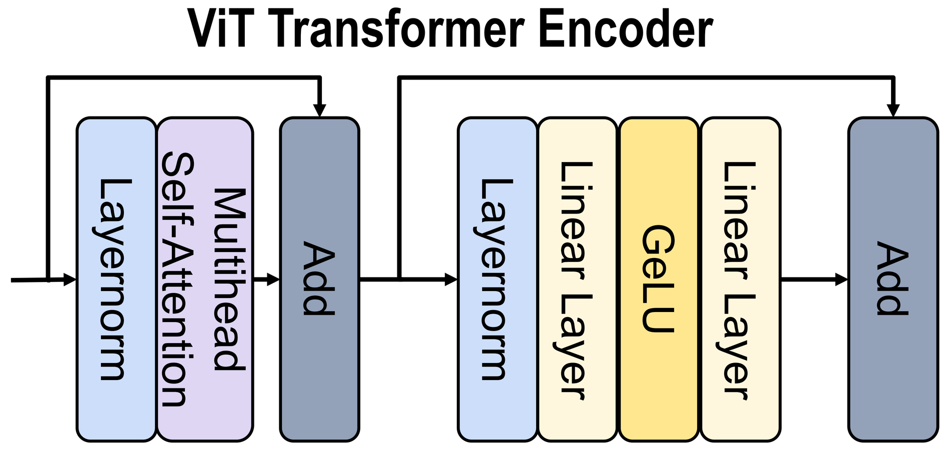
**In this work, we**
- **Developed 8-Bit self-attention kernels**
- **Implemented a tiny transformer on MCU**
- **Demonstrate an end-to-end use case**

## Vision Transformer

Transformers adapted for Computer Vision
- Mixed models w/ CNN + Transformer [1]
- **Only encoder, no decoder**
- State-of-the-art performance on ImageNet

**Promising architecture for edge application**

**ViT Transformer Encoder**




## Efficient Self-Attention Kernels

Transpositions and softmax are major bottlenecks in self-attention
- **73 % of inference latency in SpAtten [2]**
- Transpositions are impossible to parallelize on MCUs
- Softmax activation is sequence-dependent

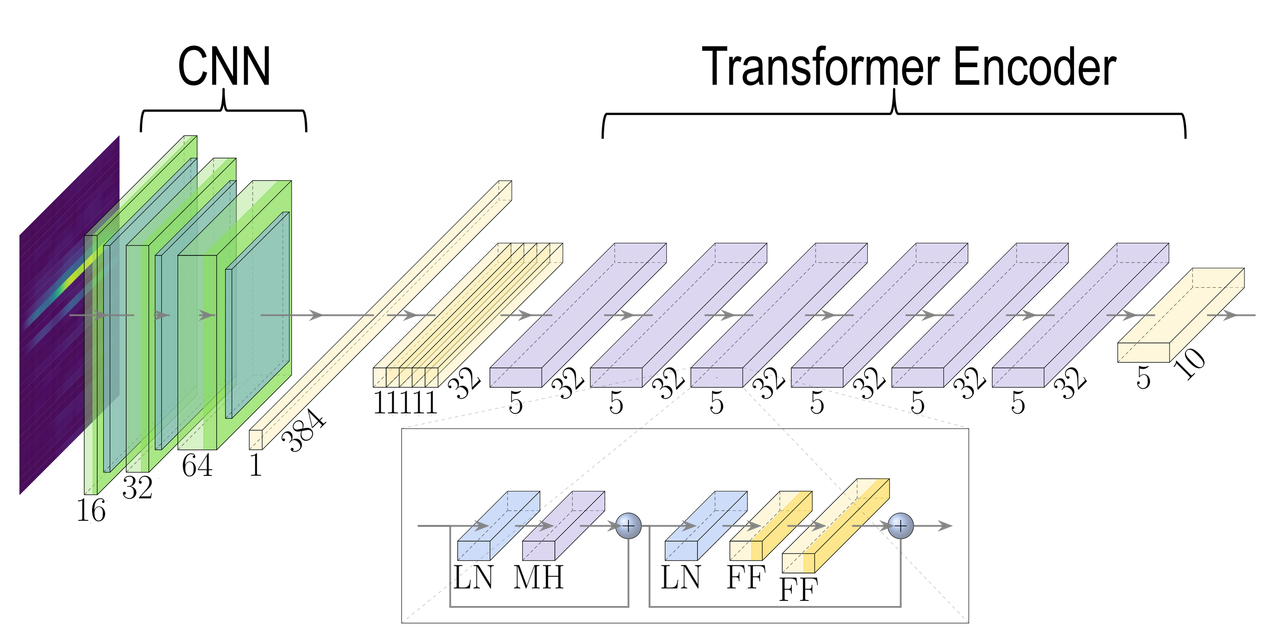
**First key idea: Fully quantize everything to 8 Bits**
- Including activations & softmax
- Leverage SIMD instructions
- 8 Bit quantization doesn't impact accuracy [3]

**Second key idea: Introduce set of specialized kernels**
- Merge softmax with matmul kernel, use quantized softmax activation [3]
- Merge transpositions with linear layer kernels by transposing data access on input matrices
- Parallelize over head dimension

**Third key idea: Parallelize over appropriate dimensions**
- Instruction parallelism: innermost loop ➡ Vector-products
- Thread parallelism: outermost loop ➡ Heads

## TinyRadar Transformer

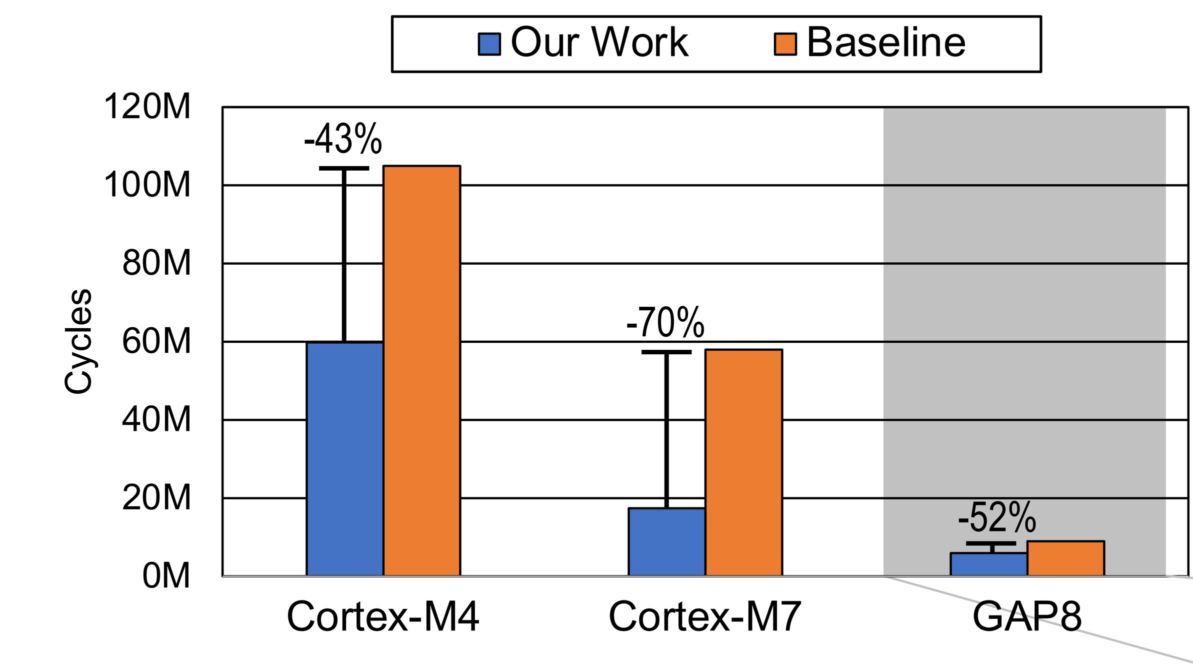
Modified version of CNN-TCN network [4] for gesture recognition
- Replaced dense convolutions by depthwise-separable convolutions
- **Replaced TCN-layers by ViT-style Transformer encoder**
- Downsampled the input data by a factor of 2 x




➡ 3 x Memory increase
➡ 3.5 % Accuracy increase, **9.6 x Latency decrease**

## Self-Attention Kernel Results

**Self-Attention Layer Performance**

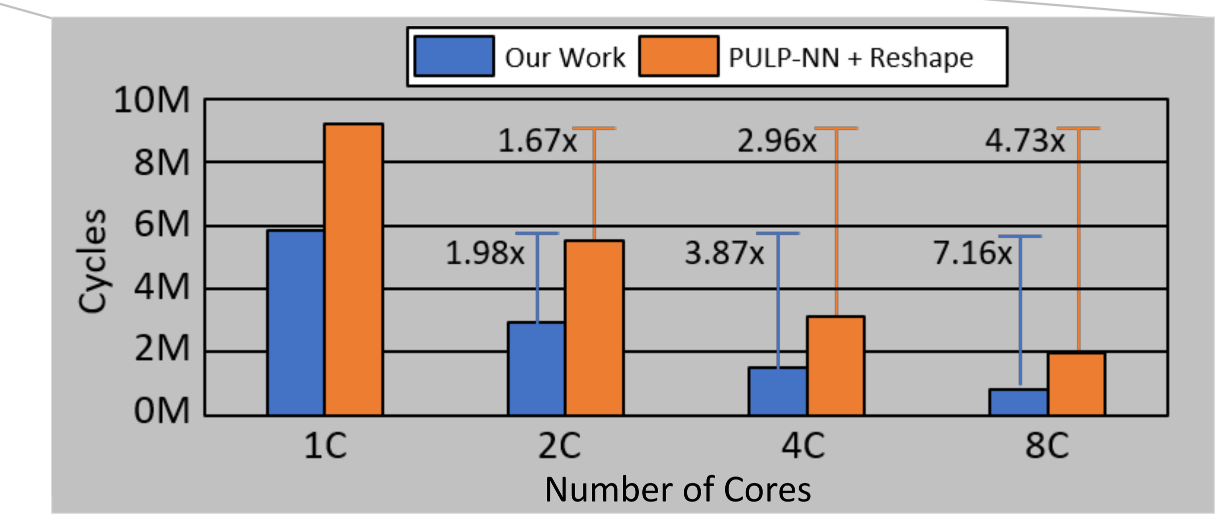



Baseline is CMSIS-NN [5] and PULP-NN [6]

Self-attention kernels reduce execution time
- 43 % on Cortex-M4
- 70 % on Cortex-M7
- 52 % on GAP8

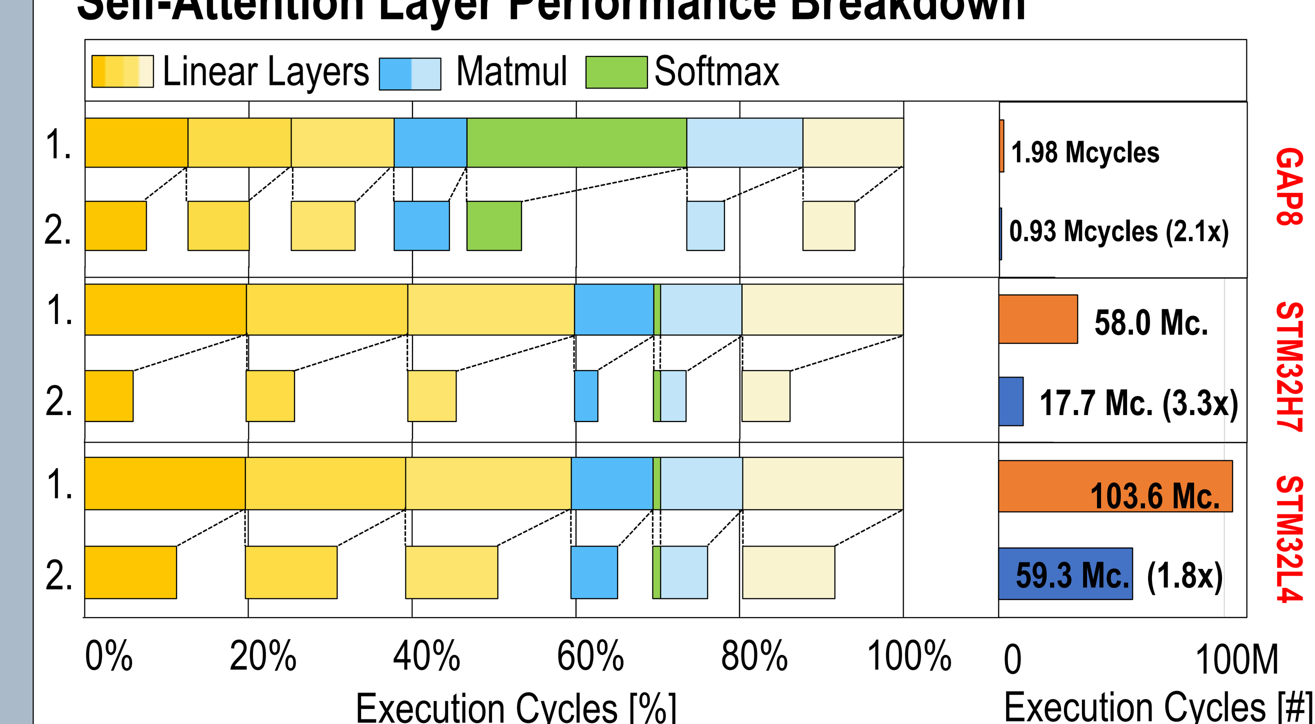
**Performance is comparable to convolutional kernels**
- 11.29 vs. 12.86 MAC/cycle on GAP8
- 0.61 vs. 0.71 MAC/cycle on Cortex-M7

Kernels parallelize more efficiently than baseline
- 1.98 x over 2 cores
- 3.87 x over 4 cores
- 7.16 x over 8 cores




**No data marshalling ➡ avoid memory bottlenecks**
**No memory bottlenecks ➡ better parallelization!**

**Self-Attention Layer Performance Breakdown**




**All speedup on single-core platforms is due to eliminating data marshalling and optimizing matmul**

Multicore performance scales further because kernels parallelize softmax

## Conclusion

In this work we presented
- Self-attention kernels with performance on-par with convolutions
  - Close-to-linear multicore scaling
- A tiny Transformer that outperforms traditional CNN/TCNs
- End-to-end results on a real-world dataset, showing
  - 3.5 % increase in Accuracy
  - **9.6 x decrease in Latency**
  - **9.6 x decrease in Energy per Inference**

The authors would like to thanks ArmaSuisse for funding this research

## References

[1]: Z. Dai et al., "CoAtNet: Marrying Convolution and Attention for All Data Sizes"
[2]: H. Wang et al., "SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning"
[3]: Kim et al., "I-BERT: Integer-only BERT Quantization"
[4]: M. Scherer et al., "TinyRadarNN: Combining Spatial and Temporal Convolutional Neural Networks for Embedded Gesture Recognition"
[5]: L. Lai et al., "CMSIS-NN: Efficient Neural Network kernels for ARM Cortex-M CPUs"
[6]: A. Garofalo et al., "PULP-NN: Accelerating Quantized Neural Networks on Parallel Ultra-Low-Power RISC-V Processors"